Uma história sobre o Software Livre/Código Aberto

Fábio Emílio Costa

Desenvolvimento de Software – Faculdades ASMEC
Rua Raul Cobra, 150, Centro – 37564-000 – Borda da Mata – MG - Brazil

fabiocosta0305@yahoo.com.br

Abstract: *This paper wants to retrospect the Free/Libre/Open Source Software (FLOSS) movement history, from its beginning (when it didn't use the term) to actual time, when it begins to make its way in the mainstream business, trying in the way to understand the potentials and flaws in this new Software Development approach and how could this empower the developer and the business by allowing partnerships between these players.*

Resumo: *Esse artigo visa traçar a história do movimento do Software Livre e de Código Aberto, de seus primórdios (quando ele não usava esse nome), até os tempos atuais, aonde ele começa a trilhar caminhos levando-o a empresas do* mainstream*, tentando no processo entender os pontos positivos e negativos desse novo modo de desenvolver software e como ele pode aumentar o poder tanto do desenvolvedor como das empresas, ao permitir a união desses em parcerias.*

1. Uma revolução silenciosa

"As revoluções de verdade, aquela que deixam marcas, em geral são silenciosas. (...) Elas não chocam e nem perturbam o mundo. Elas simplesmente vão mudando as coisas aos poucos, de maneira tão lenta que você acorda um certo dia e acaba reparando que nada é como antes." [Vaughan-Nichols 2006]

Com essas palavras, Vaughan-Nichols começou seu artigo para a Linux Watch afirmando que a Microsoft logo deveria se juntar a todos que vão percebendo que "*o código aberto não é apenas* uma forma *de desenvolver-se software, mas sim* a melhor forma*."* Na mesma semana, a Oracle viria a comprar a empresa Sleepycat, fornecedora do sistema de banco de dados aberto Berkeley DB [Newswire 2006], base de vários *software* de banco de dados, inclusive do popular SGBD MySQL, usado por empresas de alto calibre como Google e Yahoo!, todas procurando não apenas redução de custo, mas também aumento na confiabilidade e na *performance* de seus sistemas.

Além da Oracle, empresas como IBM, Sun, CISCO, Nokia, BEA e até mesmo a toda poderosa Microsoft vem procurando se envolver com a comunidade do software livre/código aberta, uma ampla comunidade de desenvolvedores de alto nível, *hackers* de computador (em um sentido diferente do difundido atualmente, e que veremos mais a diante), estudantes, pesquisadores, e toda uma comunidade que possui grandes conhecimentos em informática e desenvolve softwares de altíssimo calibre, como o Apache, que roda em 67,11% dos servidores da Internet, segundo a Netcraft [Netcraft 2006]; o PHP, que tem sido a base para uma grande quantidade de websites; e o GNU/Linux, sistema operacional que vêm se tornando o principal expoente do software livre/código aberto (que chamaremos de software livre nesse documento), enfrentando a

Microsoft com uma solução simples, segura e eficiente para servidores e, cada vez mais, para o Desktop. Iniciativas em todo o mundo, vindo de países tão díspares em economia, acesso à Internet e desenvolvimento tecnológico como Índia, Coréia, Brasil e Estados Unidos vêm vendo no software livre uma forma de reduzir não apenas os custos de suas bases instaladas e uma solução para a questão da pirataria de software, mas também uma forma de reduzir sua dependência do software proprietário, aonde uma empresa oferece apenas uma licença do software, limitando as possibilidades de "conhecer o código-fonte, de copiá-lo, redistribuí-lo ou alterá-lo, conforme as necessidades de uma sociedade" [Pinheiro IN Silveira (ORG.) 2003].

Para entendermos qual a importância desse movimento, dessa nova *metodologia* de desenvolvimento de software que vêem se delineando, precisamos entender o processo histórico de como o software vem sendo desenvolvido desde seus primórdios, de maneira a compreender e apreciar as vantagens que podem derivar do uso e desenvolvimento em software livre, ao mesmo tempo que procuramos soluções para as desvantagens.

2. Os *hackers* iniciam a revolução

Nos primórdios da informática, não existiam computadores pessoais, e sim grandes sistemas computacionais centralizados. Esses *mainframes* eram extremamente caros e complexos de serem operados, e o desenvolvimento de software era em geral específico para cada empresa ou universidade usuária, feito *inhouse*. Como afirma Taurion, "*os softwares eram gratuitos e livremente distribuídos em formato fonte (código-fonte), pois havia poucos computadores, e o valor real estava na própria máquina, e não nos programas. Vender software era algo inimaginável.*" [Taurion 2006]

Talvez pela própria quantidade de computadores, não havia mercado para a venda do software. Para suprir a necessidade por software de boa qualidade, uma grande comunidade de desenvolvedores de alto calibre, que se autodenominavam *hackers*, trocavam informações sobre como desenvolver código aproveitando as máquinas rudimentares do período no limite extremo. Nessa comunidade, a troca intensiva de informações, especificações, manuais e códigos era não somente possível, mas estimulada. Essa visão de um *hacker* pode ser estranha para alguns, mas é registrada por Eric S. Raymond no *Jargon File*, no primeiro significado do termo "*hacker*", como "*uma pessoa que aprecia explorar os detalhes de sistemas programáveis e levar suas capacidades ao limite, ao contrário da maioria dos usuários, que preferem aprender o mínimo necessário.*" [Hacker 2006].

Com o tempo e a popularização dos computadores, começou-se a popularizar-se a idéia da distribuição do software apenas de maneira *binária*, ou seja, apenas na forma das seqüências de 0s e 1s necessários para o computador executar as funções que o programador deseja, e não nas instruções legíveis pelo ser humano, o chamado *código-fonte*. Passou-se a cobrar por cada cópia do software que fosse entregue junto com os computadores e/ou vendida em separado e a licenciar tais software de modo a impedir que cópias desses binários fossem divulgadas irrestritamente, ao que chamou-se de software proprietário. Bill Gates, na época dono de uma pequena empresa de software sediada em Albuquerque, chamada Micro-Soft, chegou a enviar, em 3 de Fevereiro de 1976, uma carta ao hobbistas do *Homebrew Computer Club* (arquivada pela DigiBarn Newsletter [Digibarn 2006]), criticando essa atitude *hacker* de oferecer cópias do software uns aos outros, chegando a questiona, logo no princípio da mesma, "*se a*

comunidade hobbista seria capaz de desenvolver bom software", afirmando que o desenvolvedor "*estava sendo roubado*" e chegando a questionar "*qual hobbista iria dedicar 3 homens-ano para desenvolver um programa, encontrar todos os bugs nele, documentá-lo, e o dar o mesmo de graça*".

A resposta seria dada 5 anos depois, no MIT. Por um homem.

Richard Stallman trabalhava na época no MIT, no laboratório de Inteligência Artificial, desenvolvendo software para várias plataformas da época, principalmente o Unix. Ele é um programador de nível apuradíssimo, que na época já tinha contribuído em muitos projetos e desenvolvendo outros tantos. Em 1977, pouco tempo depois de Bill Gates extravasar sua raiva contra os hobbistas ladrões, Stallman estava esperando a impressão de um trabalho enorme de 50 páginas em seu terminal no AI Labs em uma impressora *laser* novinha dada "gentilmente" pela Xerox. Ele esperou por muito tempo, até que decidiu ir ver o que diabos estava acontecendo. O papel havia atolado.

Stallman desatolou a impressora e terminou seu trabalho. Ruminando, percebeu que seria uma enorme perda de tempo se, cada vez que a impressora resolvesse emperrar, todo mundo passasse duas horas esperando até que alguém tivesse a idéia de ir ver se a impressora estava emperrada e a desemperrar. Sendo um *hacker*, decidiu que era hora de, como Raymond definiria mais tarde em sua obra emblemática "*The Cathedral and the Bazaar*" [Raymond 2000], "coçar a sua sarna".

Stallman pensou em codificar algo simples: apenas um pequeno aviso no terminal que dissesse aos usuários: "*A impressora emperrou! Favor a desemperrar!*" Nada demais, apenas um alerta que "*avisasse a todos com trabalhos na impressora para irem consertar a impressora*" [Stallman 2002], sempre tendo a chance de ir muitas pessoas, e que uma delas soubesse o que fazewr. O problema é que a Xerox instalara apenas binários no servidor, sem fontes para que Stallman pudesse fazer sua correção. Claro que ele imaginava na época que uma empresa do porte da Xerox não iria recusar a ajuda de um programador de alto calibre do MIT capaz de oferecer um recurso interessante à sua impressora, cedendo esse código como era praxe na época, sendo um dos motivos da doação de máquinas para as universidades. E a Xerox recusou. Stallman ficou chocado.

Pouco tempo depois, ele descobriu que o pessoal do laboratório de ciências da computação da Universidade Carnegie-Mellon (CMU) tinha a mesma impressora e tinha acesso ao código fonte. Segundo a filosofia *hacker*, ele achava que a CMU não recusaria uma cópia do código-fonte a um amigo de outra instituição para que este resolvesse um problema que poderia ser de ambos. A CMU acabou recusando, pois tinham assinado um acordo de não-divulgação do fonte. Isso tinha deixado Stallman abalado, vendo a sadia cultura *hacker* desmoronar, dando lugar ao mesmo tipo de capitalismo selvagem de outras áreas. Como ele viria a afirmar: "*Esse foi o meu primeiro encontro com acordos de não-divulgação, e eu fui vítima dele. Eu e todo o laboratório fomos vítimas. E a lição que ficou é que todo acordo de não-divulgação gera vítimas.*" [Stallman 2002]

3. Uma longa empreitada

Em 1984, Stallman percebeu que essa dicotomia entre a filosofia *hacker* e o desenvolvimento fechado estava ficando cada vez maior. Como ele conta em sua biografia "*Free as in Freedom*" [Willians 2002] e em vários de seus artigos inclusos em

"*Free Software, Free Society*" [Stallman 2002], ele ficou em um dilema moral: ele poderia fazer muito dinheiro vendendo software proprietário, mas ele próprio renegando a filosofia que defendia; ele poderia deixar o ramo de informática, mas jogar fora todas as habilidades e o talento desenvolvido por ele; ou ele poderia então começar a escrever software segundo a ética que ele defendera.

Havia um porém: o próprio MIT estava começando a trabalhar com o desenvolvimento de software proprietário, que Stallman considerava anti-ético e até imoral. Ele não poderia continuar aceitando aquele fato, pois seria hipócrita da parte dele. Ele então decidiu que era hora de, como se diz no popular, "tirar o time de campo": pediu demissão do MIT e começou sua caminhada para o software livre.

A primeira coisa que percebeu foi de que precisaria de um sistema operacional, um programa que controla as operações dos programas do usuário e do computador. A opção lógica era o Unix, que Stallman mais usava na época. Porém, ele achava inaceitável usar um sistema operacional que tinha restrições que o impedisse de ver os fontes e os compartilhar com outros, como era o Unix na época (a AT&T começara a exigir acordos de não divulgação do código do Unix). Desse modo, ele não poderia, ao menos por muito tempo, utilizar qualquer sistema desses.

Mas Stallman sabia que era um programador de alto calibre. Então, ele decidiu que iria construir um sistema similar ao Unix, sem códigos proprietários, totalmente do zero. Ele chamou o projeto de GNU, um acrônimo recursivo (ou seja, o próprio acrônimo é parte do mesmo) que quer dizer *GNU's not Unix* (GNU não é Unix). Como bom *hacker*, ele gritou por ajuda para a comunidade, como mostrado no documento "*Initial Announcement of GNU Project*" [GNU 1983]. Como Stallman já era respeitado pela comunidade como um programador de elite, graças a projetos como o GNU EMACS (um poderoso editor de textos), ele foi recebendo ajuda, na forma de equipamentos, código, documentação e dinheiro, e aos poucos foi minando cada um dos milhares de pequenos componentes do Unix com suas versões livres, junto com outros desenvolvedores.

Para sustentar-se, já que Stallman estava desempregado, ele recorreu a um artifício que seria usado pelas distribuições do Linux: apesar de os programas GNU serem livres, Stallman vendia fitas com cópias dos programas e documentação por US$ 100,00 cada [Willians 2002]. Apesar desse valor aparentar absurdo atualmente, na época os custos em telefonia para transmitir tais programas tornavam a opção de comprar uma fita algo bastante interessante. Além disso, cada comprador das fitas podia, ele próprio, vender cópias, cedê-las ou disponibilizâ-las em servidores sem precisar informar a Stallman, conforme suas necessidades. Com o tempo, o artifício manteve Stallman, que, se não ficou rico, também não precisou abrir mão de sua filosofia para sustentar-se.

Aos poucos, as ferramentas GNU foram se provando melhores que as suas irmãs proprietárias, uma vez que, além de recriar suas funcionalidades originais, colocavam novas funcionalidades ou corrigiam falhas. Em muitos ambientes Unix da época, após a instalação dos sistemas proprietários costumava-se compilar o pacote de ferramentas GNU para substituir essas ferramentas proprietárias. Softwares como o GNU C Compiler (GCC) e o EMACS, entre outros, foram substituindo seus similares. Com a melhoria na qualidade e velocidade das comunicações via Internet (ainda rudimentares na época), o projeto GNU começou a receber cada vez mais contribuições de códigos e

documentação, e a *Free Software Foundation*, sua “empresa guarda-chuva”, angariava fundos que poderiam sustentar programadores que, como Stallman, estavam optando pelo desenvolvimento de software livre.

Para impedir que as contribuições da comunidade fossem tomadas por algum indivíduo ou empresa oportunista, Stallman criou a Licença Pública Geral, ou GPL. Essa licença é também conhecida como licença de *copyleft*, uma piada idiomática com o *copyright que* procura indicar a realidade: a GPL autoriza a cópia dos códigos e o uso e adaptação dos mesmos à vontade. Porém, diferentemente do que se imaginaria de imediato, isso não quer dizer que o software está em domínio público, sem dono: há uma licença de uso, ainda que permissiva, e um dono, que abria mão de alguns direitos por meio da licença. E como toda licença ela possui condições, sendo que a primordial é que todo programa que utilizar parcelas de código GPL também deverá ser licenciado segundo a GPL. Isso impede que um espertinho pegue trechos interessantes de código, coloque em seu próprio código e o feche segundo uma licença proprietária, ou que pegue um programa livre, o altere e o feche, tornando sua propriedade o que a comunidade criou.

A filosofia do software livre é definida pelo conceito de quatro liberdades. Essas quatro liberdades definem o que um programa precisa permitir para ser considerado livre. Elas são: liberdade de usar o programa para qualquer fim; liberdade de estudar o programa e adaptá-lo às suas necessidades; liberdade de melhorar o programa, corrigindo defeitos ou adicionando funcionalidades; e liberdade de redistribuir cópias, alteradas ou não, para outras pessoas. Em geral, essas liberdades só são alcançadas quando o usuário tem em mãos o código fonte. O usuário, mesmo sem saber programar, pode exercer essas liberdades, uma vez que ele pode repassar o código para um programador e indicar quais alterações serão feitas. Uma vez que o código está livre, ele pode utilizar essas alterações e repassá-las adiante. O programador, por sua vez, pode reaproveitar seu serviço em outras situações de maneira mais simples, uma vez que o código está com ele.

Como dito anteriormente, rapidamente o projeto GNU foi substituindo as ferramentas originais dos Unix proprietários por ferramentas livres. Porém, o surgimento do IBM PC e o domínio do DOS da Microsoft no mesmo estava praticamente enterrando o projeto. Além do mais, faltava uma parte importantíssima do sistema para ter-se um sistema livre completo: o *kernel*, a base do sistema operacional, o coração de todos os processos do Unix, o que poderia condenar o GNU ao fracasso. Mas o que poderia ser o cravo na sepultura do GNU foi um dos potencializadores do software livre.

4. Surge o pingüim

Da mesma forma que no caso do GNU, a história do Linux começa com uma pessoa. No caso, Linus Torvalds, na época um graduando em ciências da computação na Universidade de Helsinque, na Finlândia.

Linus precisava de um sistema Unix poderoso para realizar seus trabalhos de graduação. Porém, as estações capazes de rodar Unix naquela época custavam dezenas de milhares de dólares, e tudo o que ele podia comprar era um 386, o que tornava o Unix, na época, inviável: as variantes do Unix para PCs eram apenas brinquedos quando comparados com os sistemas Unix para sistemas Sun ou HP. A melhor opção para

Linus na época era o Minix, criado por Andrew Tanenbaum. Esse sistema possuia o código fonte aberto, mas não era livre, pois não era permitido usar o Minix para fins não-educacionais.

Linus então decidiu começar seu próprio sistema operacional: utilizando tudo o que pode aprender sobre o desenvolvimento de *kernels* para sistemas Unix no Minix, ele começou a desenvolver um novo *kernel*, utilizando as ferramentas GNU. O desenvolvimento foi muito lento, até que, como Stallman, em Setembro de 1991, Linus gritou por socorro. Mandando uma mensagem em um grupo de usuários de Minix, Linus pedia ajuda, informando que tinha construindo um *kernel* que era capaz de rodar várias das ferramentas GNU e precisava de colaboração.

Da mesma maneira que Stallman, talvez por razões diferentes (Linus afirmava que Linux era um "projeto de estimação"), Linus procurou pela ajuda da comunidade *hacker* da Internet, já muito popular e confiável. Licenciando o código do Linux segundo a GPL, Linus demonstrou que estava disposto a confiar na comunidade e que esta poderia confiar nele, o que ajudou a potencializar o projeto.

Nessa época, o projeto GNU estava trabalhando em um *kernel* baseado em uma tecnologia do MIT/CMU chamada *Mach*, o HURD. Porém, o HURD não tinha alcançando um estado no qual poderia ser considerado "usável". O Linux tinha chegado na frente, alcançando sua primeira versão considerada "usável" (0.03), ainda em outubro de 1991 [Campos 2001]. Surgia o *kernel* que o GNU procurava: surgia o GNU/Linux, ou Linux, como muitos vieram a chamar, apesar de Linux ser apenas o *kernel*.

A principal diferença entre as visões de Torvalds para o Linux e Stallman para o GNU ressaltam a principal diferença envolvendo o Software Livre e o Código Aberto. Stallman foi guiado desde o princípio por fatores de filosofia pessoal, além de uma grande necessidade de reafirmar a cultura de contribuição dos *hackers* originais. Já Torvalds fez o Linux "por diversão". Como Torvalds viria a afirmar em entrevista na finada Revista do Linux: "*Eu trabalho com código aberto porque acho que é muito mais divertido dessa maneira e, se tenho uma filosofia, ela é "quero desfrutar o que faço"*." [Conectiva 2001] Essa talvez seja a principal diferença entre ambos os grandes nomes dessa revolução silenciosa: um preocupado com questões sociais, o outro apenas preocupado com as qualidades técnicas e com a "diversão" que tal desenvolvimento pode lhe oferecer. Mas ambos causando uma revolução silenciosa.

A participação de uma comunidade multinacional, ligada via internet e com interesse no desenvolvimento do Linux foi fundamental. Essa comunidade possui participantes renomados e todos os tipos, seja pessoas como Alan Cox ou o brasileiro Marcelo Tosatti, que programam diretamente no *kernel*, ou pessoas como Alfredo Kojima, Miguel de Icaza, Guido Van Rossum ou Rasmus Lerdorf, que passaram a criar software que pudesse ser associado ao Linux. Com o tempo, esse desenvolvimento chamou a atenção de grandes corporações, como Novell e IBM, que passaram a contribuir, segundo as regras do software livre, para o crescimento do Linux.

Aos poucos, a combinação das ferramentas GNU com o *kernel* Linux foram dominando os espaços na Internet, somando-se a eles ferramentas como o servidor web Apache, as linguagens de programação PHP, Perl e Python e o banco de dados MySQL, entre outras. Projetos científicos começaram a rodar Linux para aproveitar computadores antigos em supercomputação, através de uma tecnologia livre conhecida

como *clusters* Beowulf. O Linux chegou a amadurecer ao ponto de participar da produção de filmes como Titanic. Distribuições como Red Hat, Conectiva e Mandrake (atual Mandriva), Debian e derivados (como Ubuntu, Knoppix e Kurumin) e Slackware auxiliaram a instalação do mesmo para o usuário mais leigo.

5. Ressurgindo pelo Software Livre

Enquanto o software livre dava seus primeiros passos para o *mainstream*, uma grande batalha ocorria pela dominação da Internet, agora aberta ao grande público e representando um filão potencialmente lucrativo nos negócios com informática. De um lado, a Netscape, empresa especializada em desenvolvimento de software para a Internet, que tinha criado seu navegador Netscape Navigator a partir do código do primeiro navegador para a WWW, chamado *NCSA Mosaic*, licenciado segundo a licença BSD (mais permissiva que a GPL, permitindo o fechamento do código alterado). Do outro, a toda poderosa Microsoft. Essa batalha estava sendo uma batalha que a Microsoft poderia perder facilmente: perdendo o trem da Internet, seus sistemas Windows não estavam preparados para inter-operarem com a Internet. Seria uma questão de tempo para que a Netscape minasse qualquer resistência no mundo Windows, por seus sistemas interoperarem bem com toda a Internet. Foi quando a Microsoft decidiu jogar pesado e, dizem alguns, sujo.

Comprando o *Spyglass Mosaic*, um navegador Internet razoavelmente primitivo quando comparado com o Netscape Navigator, e juntando peças diversas compradas de diversos fornecedores, a Microsoft criou o Internet Explorer e começou a distribuí-lo gratuitamente (mas não de maneira livre). Usando seu poderio de mercado, e manobras como a criação de *tags* que eram interpretadas apenas no Internet Explorer e a fusão entre o Internet Explorer e o gerenciador de arquivos Windows Explorer, rapidamente a Microsoft foi minando a Netscape do mercado. Nesse meio tempo, Eric Hahn, CTO e vice-presidente da Netscape, leu o livro "*The Cathedral and The Bazaar*" (A Catedral e o Bazar), escrito por Eric S. Raymond e publicado pela O'Reilly e em seu site na Internet [Raymond 2000].

Neste livro, Raymond diferencia a filosofia de desenvolvimento de software padrão (A Catedral), aonde um pequeno grupo de desenvolvedores criam os programas sem observar diretamente as necessidades dos usuários, lançando versões espaçadas no tempo e com pouco acesso a versões de teste, e a filosofia envolta no software livre (O Bazar), aonde uma grande quantidade de desenvolvedores criam software conforme suas necessidades pessoais ou dos usuários da comunidade, que vão testando o software conforme ele vai sendo construído. Embora isso possa significar um (teórico) maior número de falhas, já que muitos estão programando o mesmo software, isso também quer dizer que o software terá suas falhas mais rapidamente localizadas, replicadas e sanadas. Hahn decidiu que era uma boa idéia "ir ao bazar".

Em 4 de Fevereiro de 1998, conforme Raymond conta em seu livro [Raymond 2000], ele viajou até a sede da Netscape para ajudar eles a delinearem uma forma de aproveitar o Bazar. Foi criada pela Netscape uma nova licença, chamada de Licença Pública do Mozilla (MPL), bastante similar à GPL, com uma exceção: ela permitia que a Netscape utilizasse as contribuições da comunidade sem em teoria retornar nada em troca. Isso fez com que muitos não considerassem a licença "totalmente livre".

O objetivo da Netscape era claro: aproveitar-se de uma comunidade interessada

em desenvolvimento para fazer seu Netscape tornar-se muito mais poderoso que o Internet Explorer da Microsoft. Para a Netscape como empresa, essa foi uma decisão audaciosa, mas pouco efetiva: a comunidade precisou de algum tempo para entender os códigos proprietários desenvolvidos pela Netscape e pouco contribui nesse meio tempo, o que acabou enterrando essa idéia. A Netscape viria a ser dividida e vendida para a Sun e para a AOL/TimeWarner, antes de ver uma versão aberta do Netscape tornar-se produtiva, o que viria a acontecer em Novembro de 2000. Como um dos principais desenvolvedores da Netscape seria citado por Raymond, "*Software Livre não é pó de pirlimpimpim*". [Raymond 2000] Porém, o Netscape, apesar de morto, viria a ressurgir na forma do projeto Mozilla, que gerou vários produtos de software livre interessantes, licenciados pela GPL, entre os quais destacam-se o navegador *Firefox*, o editor HTML NVU e o leitor de emails *Thunderbird*.

Embora não tenha sido útil para a Netscape, o fato de abrir o seu código foi uma forma de atiçar a curiosidade dos *hackers*, e serviu para manter vivo o Netscape, e também ofereceu as bases para a comprovação de parcerias interessantes entre a comunidade e as empresas. Além disso, embora tardiamente, a idéia da Netscape provou-se presciente: atualmente, o Firefox já tomou 12% do mercado do Internet Explorer, sendo que dois em cada cinco computadores europeus usam o navegador livre [VNUnet 2006], e o Thunderbird vem sendo considerado uma alternativa segura para leitura de *email* se comparada com o seu principal rival proprietário, o Microsoft Outlook Express.

6. Uma parceria que deu certo

A iniciativa da Netscape veio a ser copiada por uma outra grande empresa em tempos recentes, gerando talvez um dos softwares com maior potencial para começar a minar a hegemonia da Microsoft.

Nos primórdios do Linux, a edição de textos era algo problemático. Embora houvessem editores de texto poderosos, como o `vi` e o EMACS, e geradores de documentos muito potentes, como o TeX e o DocBook, havia uma necessidade incrível de uma boa *suite* Office. Havia (e há) soluções livres, como o Abiword, o GNUmeric e o KOffice, mas a principal *suite* em Linux não era livre: o StarOffice, da alemã StarDivision AG. Essa *suite*, escrita em Java, rodava em multiplataforma (Windows, Linux, FreeBSD e MacOS) e oferecia um pacote que, se não estava no nível do Microsoft Office, era suficientemente boa para satisfazer as necessidades básicas da comunidade. Ela era gratuíta apenas para uso pessoal, embora suas licenças fossem mais baratas do que as do MS Office. O que gerou uma base usuária bastante intensa, que chamou a atenção de empresas que viam nessa base um mercado lucrativo.

Em 1999, a Sun Microsystems, criadora da plataforma Java, comprou a Star Division AG, por mais de US$ 70 milhões [StarOffice 2006]. Vendo na *suite* um bom potencial, a Sun resolveu realizar a mesma aposta que a já feita pela Netscape: em 19 de Julho de 2000, abriu o código do StarOffice, sob o nome de OpenOffice.org, segundo uma licença aberta, mas não livre (Licença de Fontes Padrão para a Indústria da Sun, a SISSL) [OpenOffice.org 2006].

Da mesma maneira que no caso da Netscape, a comunidade que se formou ao redor do OpenOffice.org demorou algum tempo até compreender todo aquele código monstruoso que foi cedido pela Sun (em sua versão mais atual, o código fonte da versão

estável do OpenOffice.org possui mais de 150 Megabytes, compactado). Da mesma maneira que no caso Netscape/Mozilla, levou muito tempo até que o OpenOffice.org gerasse uma versão produtiva (no caso, em 1° de Maio de 2002). A grande diferença, porém, é que a Sun, diferentemente da Netscape, podia esperar e colher os louros.

Em Maio de 2002, o OpenOffice.org 1.0 resultou no sucesso que todos os envolvidos desejavam. Seguindo uma licença dual (SISSL e LGPL) [Wikipedia 2006], ele começou a se firmar rapidamente, aproveitando, da mesma maneira que o já citado projeto Mozilla, o momento de expansão do GNU/Linux, oferecendo uma *suite* de qualidade que pudesse receber contribuições de muitas empresas e de outros projetos, como o KOffice e o Abiword, que por sua vez viriam a ser beneficiados com código que poderia ser incorporado a seus projetos. A Sun, por sua vez, ganhava com um desenvolvimento de altíssimo nível que viria a incorporar na versão 7.0 do StarOffice.

Para o usuário, as vantagens ficaram evidentes: se pudesse viver sem um suporte técnico oficial (como no caso de usuários caseiros e empresas com bons programadores), o OpenOffice.org mostrava-se uma suite muito poderosa e madura, ainda mais em sua versão 2.0, lançada em 20 de Outubro de 2005 [OpenOffice.org 2006] e com uma vasta comunidade de usuários que poderia oferecer suporte, informações, *cliparts* (como os do projeto OpenCliparts), modelos e documentações. Para o usuário que precisasse de maiores "garantias", como um suporte técnico especializado e uma garantia legal, o StarOffice oferecido pela Sun a um custo razoável oferecia todas as necessidades de uma *suite* poderosa e rápida.

A prova da velocidade da comunidade viria em 2004, com o surgimento do *OpenDocument Format* (ODF), um padrão aberto para arquivos Office criados pela OASIS, uma união de empresas que, entre outras, envolvia a Sun e a Microsoft. Assim que as primeiras versões *draft* (ou seja, não-oficiais) foram divulgadas na Internet, os *hackers* e desenvolvedores começaram a escrever código, dentro do OpenOffice.org 1.9.49 (uma das primeiras versões *beta* do OpenOffice.org 2.0), para abrir arquivos segundo os padrões da OASIS. Assim que o padrão foi liberado em sua versão 1.0, a comunidade do OpenOffice.org, em sua versão 1.9.109 (*2.0 Release Candidate 1*) conseguiu criar o primeiro software com suporte ao ODF, seguida pela Sun com o StarOffice 8.0 (obviamente), e pelos projetos que coopetem (ou seja, cooperam e competem) com ele, o KOffice, o AbiWord e o GNUmeric.

Com tantas opções, os formatos ODF começaram a se tornarem atrativos, ainda mais depois que o Estado norte-americano de Massachussets passou uma resolução para aceitar documentos apenas em formatos que fossem abertos (que possuam para a criação de leitores e conversores independentes), como os ODF e o formato PDF da Adobe . Outras empresas, como a Corel, posicionaram-se a favor da iniciativa do ODF, seja para edição, pesquisa, ou visualização. E claro, sempre existirá o OpenOffice.org como opção.

7. Conclusão

A revolução silenciosa está começando a fazer barulho. Depois de comprovar-se uma iniciativa atrativa no ambiente de servidores Web e de começar uma penetração ousada em servidores de missão crítica, o software livre começa a penetrar pelo que Eric S. Raymond viria a conceituar como a "linha Maginot da TI". Se existe uma "linha Maginot da TI", então software como o GNU/Linux, os software do projeto Mozilla e o

OpenOffice.org estão a contornando, como os alemães contornaram sua irmã bélica.

Entretanto, é importante notar que, assim como conquistas, isso vem causando desafios à comunidade do software livre: originalmente, a comunidade se movia para os projetos considerados *sexy,* ou seja, que permitiriam maior visibilidade aos desenvolvedores e/ou ofereciam maiores atrativos técnicos, seja pela qualidade do código ou pelo desafio oferecido. Agora, uma cada vez maior base de usuários leigos está se formando ao redor desses software. Usuários que poderão ser vistos como os "trouxas" de Harry Potter, incapazes de compreenderem a "magia" dos "bruxos" da comunidade do software livre. Dicotomia essa que foi registrada por Eric S. Raymond em seu *Jargon File*, com os termos *wizard* (bruxo) e *muggle* (trouxa) [Jargon 2006].

As vantagens técnicas devem ter ficado claras neste documento, seja para o desenvolvedor, para as empresas e para os usuários. Mas é importante amadurecer ainda mais o software: existem muitos outros projetos interessantes, como o GIMP (editor de imagens similar ao Photoshop), o GAIM (um programa de mensagens instantâneas multi-redes), e o já citado NVU, que, embora sejam de altíssimo nível, ainda precisam amadurecer para serem capazes de enfrentar seus "rivais" proprietários.

Além disso, é necessário lidar com os egos dos desenvolvedores, evitando que as "guerras santas" tirem o foco do desenvolvimento pro-ativo para o usuário final. Iniciativas como ODF e FreeDesktop.org estão mostrando o caminho, e a troca de códigos entre projetos livres "rivais" começa a se tornar uma realidade e algo atrativo para todas as partes, como comprovado no caso do OpenOffice.org contra o AbiWord e o KOffice.

A consciência de que o software pode ser adaptado às necessidades de cada um, algo originalmente inerente aos usuários de computador, está ressurgindo aos poucos. As centenas de distribuições GNU/Linux, e os milhares de projetos no Sourceforge.Net são a prova disso. O software livre passa por uma fase maravilhosa que, se bem aproveitada e se evitando-se os riscos apresentados nesse documento, poderá ser a resposta à pergunta que Bill Gates fez em 1976. Será a prova de que Richard Stallman estava certo, de que Eric S. Raymond foi presciente ao ver o bazar e de que Linus Torvalds se divertiu muito mais ao chamar a comunidade.

Os desafios são muitos: as proteções anti-cópia através de sistemas de Controle de Direitos Digitais (DRM), as legislações draconianas como a Lei para o Copyright do Milênio Digital (DMCA) e os aproveitadores oportunistas que vêm ameaçando a comunidade das mais diversas maneiras. Mas o software livre tem o potencial de, mesmo com esses desafios e todos os que já são impostos por si, mesmo, oferecer uma chance de uma sociedade mais justa, incluída e desenvolvida, de maneira mais igualitária e com maiores chances de aprendizado e crescimento, com vantagens para todos em um jogo aonde todos têm a ganhar.

Essa é uma grande oportunidade de crescimento e ganhos saudáveis para todos: usuários poderão ganhar com a melhoria da qualidade do software; desenvolvedores poderão aprender e colaborar para a melhoria de uma alta gama de software; e empresas receberão software de altíssimo calibre a custo baixo. É tudo uma questão de colaboração.

Referências

[Vaughan-Nichols 2006] - VAUGHAN-NICHOLS, Steve J. (2006) - "Open Source – the one, true way to develop software" – Linux Watch, http://www.linux-watch.com/news/NS3670538334.html, publicado em 16/02/2006, acessado em 20/02/2006.

[Newswire 2006] - PR Newswire Association LLC - "*Oracle Buys Open Source Software Company Sleepycat*" - http://www.prnewswire.com/cgi-bin/stories.pl?ACCT=104&STORY= /www/story/02-14-006/0004281592&EDATE=, publicado em 14/02/2006, acessado em 20/02/2006

[Netcraft 2006] - NETCRAFT – February 2006 Web Server Survey – http://news.netcraft.com/archives/2006/02/02/february_2006_web_server_survey.html, publicado em 02/02/2006, acessado em 20/02/2006

[Pinheiro IN Silveira (ORG.) 2003] - PINHEIRO, Walter - "*A luta pelo Software Livre no Brasil*" IN DA SILVEIRA, José A. - "*Software Livre e Inclusão Digital*" - São Paulo, 2003; Conrad.

[Taurion 2006] – TAURION, Cezar - "*A história do software*" - http://noticias.uol.com.br/mundodigital/softwarelivre/2004/09/15/ult2449u3.jhtm, publicado em 15/09/2004, acessado em 20/02/2006

[Hacker 2006] - JARGON FILE - "*Hacker*" - http://www.catb.org/jargon/html/H/hacker.html, acessado em 20/02/2006.

[Digibarn 2006] - DIGIBARN - "*Homebrew Computer Club Newsletter Volume 2, Issue 1 - Extract of Bill Gates' Open Letter to Hobbyists*" - DigiBarn Newsletters – http://www.digibarn.com/collections/newsletters/homebrew/V2_01/gatesletter.html#gatesletter, acessado em 20/02/2006.

[Raymond 2000] RAYMOND, Eric S. - *"The Cathedral and the Bazaar"* - Sebastopol, USA; O'Reilly, 2000, disponível em http://www.catb.org/~esr/writings/cathedral-bazaar/, acessado em 20/02/2006

[Willians 2002] - WILLIAMS, Sam - *"Free as in Freedom - Richard Stallman's Crusade for Free Software"*; Sebastopol – USA, 2002; O'Reilly, disponível em http://www.oreilly.com/openbook/freedom/index.html, acessado em 20/02/2006

[Stallman 2002] - STALLMAN, Richard - "*Free Software, Free Society: Selected Essays of Richard M. Stallman*"; Boston – USA, 2002; GNU Press, disponível em http://www.gnupress.org/philosophy/fsfs/rms-essays.pdf, acessado em 21/02/2006

[GNU 1983] – STALLMAN, Richard - "*Initial announcement of the GNU Project*" - http://www.gnu.org/gnu/initial-announcement.html, acessado em 21/02/2006

[Campos 2001] - CAMPOS, Augusto - "*Viagem ao coração do Linux*" IN Revista do Linux, número 13, Janeiro de 2001

[Conectiva 2001] - CONECTIVA - "*Divulgar é preciso*" - Entrevista com Linus Torvalds IN Revista do Linux, número 18, Junho de 2001

[StarOffice 2006] - WIKIPEDIA - "StarOffice" - http://en.wikipedia.org/wiki/StarOffice, acessado em 21/02/2006;

[OpenOffice.org 2006] - WIKIPEDIA - "OpenOffice.org" -

http://en.wikipedia.org/wiki/OpenOffice.org, acessado em 21/02/2006;

[Jargon 2006] - JARGON FILE - http://www.catb.org/jargon/, acessado em 21/02/2006.

[VNUnet 2006] – THOMSON, Iain - "*European Firefox use hits 20 per cent*" - http://www.vnunet.com/vnunet/news/2148740/european-firefox-hits-per-cent, publicado em 18/01/2006, acessado em 25/02/2006